DOCKET NO: 290147US8

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

IN RE APPLICATION OF	:
THOMAS R. VOLPERT	: EXAMINER: MATTHEW T. HENNING
SERIAL NO: 09/888,316	:
2ND RCE FILED: MAY 2, 2006	: GROUP ART UNIT: 2131
FOR: SYSTEM AND METHOD FOR DATA ENCRYPTION	:

APPEAL BRIEF

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

This is an appeal of the Rejection of Claims 1, 3, 5-10, 21-23, and 25-62 in the Official Action of July 20, 2006. A Notice of Appeal was timely filed on August 15, 2006.

I. REAL PARTY IN INTEREST

The real party in interest in this appeal is the Assignee, Uponus Technologies, L.L.C..

II. RELATED APPEALS AND INTERFERENCES

Appellants' legal representative and the Assignee are aware of no appeals or interferences, which are believed to directly affect, be directly affected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF THE CLAIMS

Claims 1, 3, 5-10, 21-23, and 25-62 are presently active in this case. Claims 1, 3, 5-10, 21-23, and 25-62 have been twice rejected and form the basis for this appeal. The attached Claim Appendix includes a clean copy of Claim 1, 3, 5-10, 21-23, and 25-62.

IV. STATUS OF THE AMENDMENTS

No Amendment was filed subsequent to the July 20, 2006 rejection.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

A concise explanation of the inventions defined in the independent claims is now provided.

In the exemplary claimed advancement, as recited in independent Claim 1, a method of encrypting an input data string, including a plurality of bits of binary data with a processing device communicatively coupled to a memory having an encryption program is recited in the preamble of independent Claim 1. Support for the context provided by the preamble can be found at Fig. 1 (20) input data string, memory (60); Fig. 5, processing device (310), memory (320), (316), encryption program (318) and the accompanying description of these figures in the specification at page 4, line 12 through page 5 line 14, and page 19 line 15 through page 21 line 7, respectively.

Claim 1 recites a method step of receiving an input data string for encryption at the processing device. Support for this claimed feature is found at Fig. 1 (120) and Fig. 2 (110) and the corresponding descriptions in the specification at pages 4 line 12 through page 5 line 14. A further method step of providing a control index in the memory, the control index being defined prior to receiving the data input string for encryption of the processing device,

the control index including a plurality of control codes is also provided. Support for this claimed feature is found for the control code index at (60) of Fig. 1, which includes a list of control codes therein. Fig. 5 provides memory (316) and (320). Fig. 3 detects the memory and/or organization of the control codes in the control code index. The control code index being defined prior to receiving input data string is described at page 7 line 6 through page 8 line 17 and at page 13 line 1 through page 14 line 8.

Independent Claim 1 further recites determining an order in which to query the presence of each of 2^n different configurations of n bits within an input data string. Support is found for this claimed feature at Fig.2 (130) and is described at line 24, page 5 through line 2, page 6; the determined order is described at page 13, lines 5 – 23 of the specification.

Independent Claim 1 further recites a step of generating a control code associated with a determined order using the control code index, the values of the generated control code being independent of input data strings specific characteristics. Support for these claimed features is found at Fig. 1 (140), Fig. 2 (140), Fig 3 (62), control code index application at page 13, line 24 through page 14, line 7; page 18, lines 14-19 (without analyzing the input data string so as to be independent thereof).

Independent Claim 1 further recites a step of generating a position code by identifying positions of each of the 2^n different configurations of n bits in the input data string in accordance with the determined order. Support for this method step is found at Fig. 1 (150), position code generation, Fig. 2 (154), generate position code, page 8, line 18 through page 11, line 5. Independent Claim 1 recites the method step of combining the control code and the position code to form an encrypted data string. Support is found in the specification for this method step at Fig. 2 (160), combine control code, block code, and position code, and page 11, lines 6-17.

The subject matter of independent Claim 21, independent computer executable method Claim 23, and apparatus Claim 62 essentially parallel that at independent Claim 1, as fully outlined above.

The difference between Claim 21 and Claim 1 is that Claim 21 recites a different preamble of "a method for encrypting an input data string, including a plurality of bits of binary data." All other method steps of Claim 21 are substantially similar to that of Claim 1. Thus, the preamble of Claim 21 is supported by the same citations provided above with reference to the preamble of Claim 1.

Claim 23 recites a preamble of a computer executable method in which a computer-readable medium includes computer program instructions that cause a computer to implement a method of encrypting an input data string including a plurality of bits of binary data. All other method steps of Claim 23 are substantially similar to Claim 1. Support for the computer implementation of this method is found at Fig. 5 in the corresponding description in the specification, which begins at page 19, line 15 through page 21, line 7.

Claim 62 recites an apparatus for implementing the above-supported methodology. The preamble of Claim 62 recites an electronic device for encrypting an input data string including a plurality of bits of binary data. Support for the preamble is found in the electronic device illustrated in Fig. 5 and the encryption of the input data string finds support in Figs. 1-2, as noted above. Additionally, the processor and memory structure, which is incorporated with the above-noted methodology in Claim 62, is shown in Fig. 5 and the corresponding discussion in the specification of this figure, which begins on page 19, line 15 through page 21, line 7.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The first ground to be considered on appeal is whether Claims 1, 3, 5-10, 21-23, and 25-62 fail to comply with the written description requirement under 35 U.S.C. §112, first paragraph.

The second ground to be considered on appeal is whether Claims 1, 3, 5, 8-10, 21-23, 25, 26, 29-40, 44-55, and 59-62 are unpatentalble over DeMaine et al. (U.S. Patent No. 3,656,178, hereinafter DeMaine), and further in view of Cellier et al. (U.S. Patent No. 5,884,269, hereinafter Cellier) under 35 U.S.C. § 103(a).

The third ground to be considered on appeal is whether Claims 6, 7, 27, and 28 are unpatentalble over DeMaine and Cellier as applied to Claims 5 and 26, respectively, and further in view Shimizu et al. (U.S. Patent No. 6,772,343, hereinafter Shimizu) under 35 U.S.C. § 103(a).

The fourth ground to be considered on appeal is whether Claims 41, 42, 56, and 57 are unpatentalble over DeMaine and Cellier as applied to Claim 1, and further in view of Weiss (U.S. Patent No. 5,479,512) under 35 U.S.C. § 103(a).

The fifth ground to be considered on appeal is whether Claims 41, 43, 56, and 58 are unpatentalble over DeMaine and Cellier as applied to Claim 1, and further in view of Butler et al. (U.S. Patent No. 5,861,887, hereinafter Butler) under 35 U.S.C. § 103(a).

VII. ARGUMENT

First Issue

A. THE OFFICE HAS FAILED TO ESTABLISH REQUIRED *PRIMA FACIE* AND FACTUAL SHOWINGS

The July 20, 2006 Official Action rejects Claims 1, 3, 5-10, 21-23, and 25-62 as failing to comply with the written description requirement under 35 U.S.C. § 112, first paragraph. The Official Action contends that the claims recite subject matter, which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor, at the time the Application was filed, had possession of the claimed invention. Applicant respectfully appeals this rejection.

The MPEP, and, more specifically § 2163 (Rev. 3, 08/05), sets forth guidelines for all Patent and Trademark Personnel. These guidelines (at MPEP § 2163 II A) state that:

> Office personnel should adhere to the following procedures when reviewing patent applications for compliance with the written description requirement of 35 U.S.C. 112, para. 1. The examiner has the initial burden, after a thorough reading and evaluation of the content of the application, of presenting evidence or reasons why a person skilled in the art would not recognize that the written description of the invention provides support for the claims. There is a strong presumption that an adequate written description of the claimed invention is present in the specification as filed, *Wertheim*, 541 F.2d at 262, 191 USPQ at 96; however, with respect to newly added or amended claims, applicant should show support in the original disclosure for the new or amended claims. See MPEP § 714.02 and § 2163.06 ("Applicant should * * * specifically point out the support for any amendments made to the disclosure."); and MPEP §2163.04 ("**If applicant amends the claims and points out where and/or how the originally filed disclosure supports the amendment(s), and the examiner finds that the disclosure does not reasonably convey that the inventor had possession of the subject matter of the amendment at the time of the filing of the application, the examiner has the initial burden of presenting evidence or reasoning to explain why persons skilled in the art would not recognize in the disclosure a description of the invention defined by the claims**."). (Emphasis added).

Clearly, this section of the MPEP requires that the present outstanding Action must present "evidence or reasoning to explain why persons skilled in the art would not recognize" a disclosure of :

(a) "values of the generated control code being independent of input data string specific characteristics";[1]

(b) "the control code index being defined prior to receiving the input data string"[2]

upon review of applicant's disclosure and figures.

MPEP § 2163.04 (I) goes even further in requiring that when rejecting any claim on this basis, the Action:

> . . . **must set forth express findings of fact which support the lack of written description conclusion** (see MPEP § 2163 for examination guidelines pertaining to the written description requirement). These findings should:
>
> (A) Identify the claim limitation at issue; and
>
> (B) **Establish a *prima facie* case by providing reasons why a person skilled in the art at the time the application was filed would not have recognized that the inventor was in possession of the invention as claimed in view of the disclosure of the application as filed**. (Emphasis added).

While the claim features at issue have been identified above, no reasons have been presented to explain "why a person skilled in the art at the time the application was filed would not have recognized that the inventor was in possession of the invention as claimed in view of the disclosure of the application as filed." Likewise, there has been no express finding of fact setting forth which support the lack of written description conclusion. A detailed discussion of each claim limitation at issue is provided below.

---

[1] Claim language at issue (claims 1, 21, 23 and 62)
[2] Claim language at issue (claims 1, 21, 23 and 62).

1. "VALUES OF THE GENERATED CONTROL CODE BEING INDEPENDENT OF INPUT DATA STRING SPECIFIC CHARACTERISTICS"

There was no factual evidence or reasoning provided to explain why persons skilled in the art would not recognize a disclosure of "control code being independent of input data string specific characteristics" in considering at least the descriptions noted below, along with the referenced figures of the specification which describe:

> In response to the **frequency and relationship analyses, a control code can be generated** through the use of the control code index 62, depicted in Fig. 3. The control code index 62 can be used to **select the control code that indicates both the appropriate order** of the combinations of bits and the appropriate position code routine. In this example, the combinations of bits are arranged in descending order from the most to least frequent based upon the relative frequency of each of the combinations of bits. Accordingly, the control code for the present example can indicate the following order for the four different combinations 00, 10, 01, 11. The control codes corresponding to this order are found at numbers 5 and 6 in the control code index 6[3] (emphasis added)[4]

As one of skill in the art would instantly recognize from the embodiment described above, the control code may be generated in response to frequency and relationship analysis of the data.

Alternatively, as described at page 18 of the specification:

> In a further alternative embodiment, the control code can be generated **without analyzing the data** (step 130, FIG. 1). For example, the control code can be automatically generated based on a default setting whereby a specific control code is **automatically selected** from the control code index 60. Alternatively, the control code can be generated in a random or non-random fashion through the use of the control code index 60. (emphasis added)

[3] Application at pages 13-14; See also page 2 lines 14-18; page 4 lines 17-19; page 7 line 5 through page 8 line 18; page 9 lines 4-9; page 13 lines 23 through page 15; page 17 line 19 through page 18 line 1; page 18 lines 14-18 and page 20 line 21.

[4] Applicant notes that (62) of Fig. 3 refers to a selected column of the control code index (60) and (64) of Fig 4 refers to an expanded list of two and three bit pair index entries.

The relevant disclosure was clearly stated in the last response to be described throughout identified portions of the specification and figures.[5]

In response, the Official Action of July 20, 2006 provided:

> The applicant appears to have misinterpreted the rejection, in that the rejection is based on a lack of support for the limitation that the values of the control code being independent of input data string specific characteristics, as recited in the claims, and not the selection of the control code being independent.[6] (emphasis added)

The above noted statement of the Office, as well as earlier statements, fails to set forth:

1. Express findings of fact which support the lack of written description conclusion; or

2. Reasons as to why one of skill in the art at the time the application was filed would not have recognized that the inventor was in possession of the invention as claimed in view of the disclosure of the application as filed. (emphasis added)

Indeed, with regard to the statement of July 20, 2006, it appears as though it is the Office which is confusing issues. Applicant's claims recite values of **a generated control code** being independent of input data string characteristics. To be clear, the relevant portion of claim 1 is reproduced below:

> ....generating a control code associated with the determined order using the control code index, the values of the generated control code being independent of the input data string specific characteristics....(emphasis added)

Thus, in contrast to the July 20, 2006 statement, by the clear language of the claims, it is the **generated control code which is recited as having values independent of the input data string**. Thus, if a specific control code of the control code index is generated, by default, for example, the values of this generated control code are independent in that the control code is generated independent of input data string characteristics as recited in the

[5] See Response of May 2, 2006
[6] See Official Action of July 20, 2006 at page 3

claims. As noted throughout the specification, and, at least at page 8 "**Generating control codes (step 140) involves selecting a control code from the control code index 60**...." As can be appreciated, a control code is generated (i.e., selected) in an exemplary embodiment, based upon values of the control code in relation to input data characteristics[7], or, based upon a random, default or like condition, which generates a control code independent of input data.[8]

In this regard, the required factual showings and explanations necessary to present a *prima facie* case under 35 U.S.C. § 112, first paragraph, has not been provided. There has been no explanation provided by the Office as to why one of ordinary skill in the art would not recognize the above noted claimed feature in considering at least the above noted portions of the Applicant's specification. Likewise, no express findings of fact have been provided to support the Office's conclusion.

Accordingly, this rejection must therefore, be reversed.

2. "THE CONTROL CODE INDEX BEING DEFINED PRIOR TO RECEIVING THE INPUT DATA STRING"

There was no factual evidence or reasoning provided to explain why persons skilled in the art would not recognize a disclosure of "the control code index being defined prior to receiving the input data string" in considering at least the description noted below, along with the referenced figures of the specification which describe that the control code index includes individual index entries. Further details of the control code process are recited at pages 7-8 which describes, *inter alia:*

[7] Application at page 13 line 1 through page 14 line 3.
[8] Application at page 18 lines 14-18.

> The control code index 60 comprises a list of control codes that each correspond to a specific order of the combination of bits. The four combinations of two bits 00, 01, 10, 11) can be **arranged** in 24 different orders. One or more different control codes can be **assigned** to the same order of the combinations of bits as seen in the control code index 60.

As can be appreciated from the plain language above, the current inventor clearly possessed "the control code index being defined prior to receiving the input data string" Tellingly, the language of the inventor's specification utilizes language like **"assigned"** and **"arranged"** to clearly denote to a person of skill in the art that the establishment of the control code index **arrangements and assignments** may be prior to the reception of data. For example, one of ordinary skill in the art would instantly recognize that, in streaming data applications, such assignment and arrangement would be done prior to receiving the input data string to ensure real time processing of input data.

The relevant disclosure was clearly stated in the last response to be described throughout identified portions of the specification and figures.[9] In response, the Official Action of July 20, 2006 provided the cursory statement:

> ...the examiner does not find the argument persuasive. Fig 3 is merely a sample control code index and neither Fig 3 or the text of the specification regarding Fig. 3 disclose or even suggest that the control code index was available prior to the reception of the input data string.[10]

The above noted statement of the Office, as well as earlier statements, fails to set forth:

2. Express findings of fact which support the lack of written description conclusion; or

2. Reasons as to why one of skill in the art at the time the application was filed would not have recognized that the inventor was in possession of the invention as claimed in view of the disclosure of the application as filed. (emphasis added)

[9] See Response of May 2, 2006
[10] Official Action of July 20, 2006 at pages 2-3

In this regard, the required factual showings and explanations necessary to present a *prima facie* case under 35 U.S.C. § 112, first paragraph, has not been provided. There has been no explanation provided by the Office as to why one of ordinary skill in the art would not recognize the above noted claimed feature in considering at least the above noted portions of the Applicant's specification. Likewise, no express findings of fact have been provided to support the Office's conclusion.

Accordingly, this rejection must therefore, be reversed.

B. REPHRASING OF SPECIFICATION DESCRIPTION IS NOT NEW MATTER

To the extent that the outstanding Action is attempting to base this rejection on the failure of the specification to present the same words used in the noted claims ("the control code index being defined prior to receiving the input data string" or "control code being independent of input string specific characteristics), no such requirement exists. See MPEP § 2106.07 that states:

> Mere rephrasing of a passage does not constitute new matter. Accordingly, a rewording of a passage where the same meaning remains intact is permissible. *In re Anderson*, 471 F.2d 1237, 176 USPQ 331 (CCPA 1973).

Accordingly, the present rejection of Claims 1, 3, 5-10, 21-23 and 25-62 under the written description requirement of the first paragraph of 35 U.S.C. § 112 does not comply with the requirements of the MPEP and case law cited therein. The required explanations and reasons required by these authorities in support of such a rejection have not been presented.

Consequently, Applicant respectfully request that the rejection of Claims 1, 3, 5-10, 21-23, and 25-62 under 35 U.S.C. § 112, first paragraph, be reversed.

C. OBJECTIONS UNDER 37 C.F.R. 1.75(d)(1) and M.P.E.P. § 608.01(o)

The July 20, 2006 Official Action objects to the specification for failing to provide proper antecedent basis for the claimed subject matter.

As Applicant has identified support for the noted claim terminology ("the control code index being defined prior to receiving the input data string" or "control code being independent of input string specific characteristics) this objection has been addressed above. However, for purposes of completeness, Applicant notes that an "explicit" disclosure of a claim term is not required. An invention claimed need not be described *ipsis verbis* in the specification in order to satisfy the disclosure requirements. (*Ex Parte Hope*, 19 U.S.P.Q. 2d. 1211 (*Bd. Pat. App. & Inter 1991*).

Consequently, Applicant submits that this objection, although not for consideration on appeal, is, nonetheless improper.

D. OBJECTION TO CLAIMS 57-58

The July 20, 2006 Official Action objects to Claims 57 and 58 as being grammatically incorrect. Applicant will address this informality upon disposition of issues 1-5 by the board.

Second Issue

The July 20, 2006 Official Action rejects Claims 1, 3, 5, 8-10, 21-23, 25, 26, 29-40, 44-55, and 59-62 as being unpatentable over DeMaine, and further in view of Cellier under 35 U.S.C. § 103(a). The Official Action contends that De Maine discloses all of the Applicant's claim limitations, with the exception of a control code index. However, the Official Action cites Cellier as disclosing this more detailed aspect of the Applicant's invention and states that it would have been obvious to one skilled in the art at the time the invention was made to combine the cited references for arriving at the Applicant's claims. Applicant respectfully appeals this rejection.

Applicant's amended Claim1 recites, *inter alia*, a method for encrypting an input data string including a plurality of bits of binary data, including:

> . . . generating a control code associated with the determined order using the control code index, the values of the generated control code being independent of input data string specific characteristics;
> generating a position code by identifying positions of each of the 2^n different configurations of n bits in the input data string in accordance with the determined order. . .

De Maine describes four compression techniques (i) Slow Mode Type 1 compression, (ii) Slow Mode Type 2 compression, (iii) Fast Mode Type 1 compression, and (iv) Fast Mode Type 2 compression.

Turning first to the Slow Mode Type 1 compression and the Slow Mode Type 2 compression, both techniques begin with an initial analysis of the input data string. The input data string is scanned on a byte-by-byte basis.[11] A LEXICON table is provided with 256 byte positions where each byte position corresponds to one of the 256 different 8-bit configurations possible for a single byte of data. The LEXICON table is used to count the

[11] De Maine at column 91, lines 47-65.

number of times that each of the different byte configurations appear in the scanned input data string. Those byte configurations that do not appear in the scanned input data string are designed as Type 1 codes and those byte configurations that are identified as appearing more than a certain number of times in the scanned input data string are designated as Type 2 codes.[12]

The Slow Mode Type 1 compression is performed first by analyzing the input data string for the presence of redundant multi-byte patterns. The identified redundant multi-byte patterns are deleted from the input data string and replaced with a Type 1 code that was identified during the initial analysis of the input data string. Each deleted multi-byte pattern and the associated replacement Type 1 code are inserted at the beginning of the compressed data string. The Slow Type 1 compression is repeated until either all of the identified Type 1 codes have been utilized or until the process fails to achieve further compression.

The Slow Mode 2 compression is performed next on the output of the Slow Mode Type 1 compression and examines consecutive 256 byte string segments for the presence of each of the Type 2 codes identified during the initial analysis of the input data string. If a particular Type 2 code is found to appear multiple times in a string segment, a 256 bit map (32 bytes long) is generated identifying the specific locations of that Type 2 code within the 256 byte string segment. The redundant Type 2 code is deleted from the string segment and the string compressed to eliminate the spaces vacated by the deleted Type 2 code. The deleted Type 2 code and the 256 bit map are added to the compressed string segment.[13]

Both the Slow Mode Type 1 and the Slow Mode Type 2 compression techniques involve an analysis of specific characteristics of the input data string for generating Type 1

---

12 De Maine at column 92, lines 5-13.
13 De Maine at column 92, lines 31-46.

codes and Type 2 codes. More specifically, those byte configurations that are identified as not appearing in the input data string are designated Type 1 codes and those byte configurations that are identified as appearing more than a certain number of times within the input data string are designated as Type 2 codes.

In contrast, the values of the control codes in the control code index, as recited by the claims at issue, are **selected to be independent of input data string specific characteristics**. Furthermore, unlike the teachings of De Maine, where the LEXICON table defining the Type 1 codes and the Type 2 codes are generated on the fly as a component of the process, the control codes in the control code index are defined prior to even receiving the input data string for encryption.[14]

Turning now to the Fast Mode Type 1 and the Fast Mode Type 2 compression techniques, both of these compression techniques involve the creation and use of a PCORDS table. The PCORDS table is a dynamic table that is created based on the historical analysis of the characteristics of previously compressed input data strings and is updated continuously based on the input data string characteristics of every new input data string received for compression.[15]

A first section of the dynamic PCORDS table, used in Fast Mode Type 1 compression, contains a listing of multi-byte patterns that are likely to occur in similar types of input data strings and a savings ratio associated with each multi-byte pattern to indicate the degree of compression achieved by the use of that multi-byte pattern. During Fast Mode of Type 1 compression, the received input data string is analyzed for the presence of each of the multi-byte patterns identified in the PCORDS table and the PCORDS table is dynamically

[14] Specification at Fig. 3.
[15] De Maine at column 96, lines 66-69.

updated to reflect the likelihood that the multi-byte patterns actually identified as being present in the received input data string are likely to occur in future input data strings.

A second section of the PCORDS table, used in Fast Mode Type 2 compression, contains a listing of Type 2 codes that are likely to occur in similar types of input data strings. During Fast Mode Type 2 compression, each string segment is analyzed for the presence of each of the Type 2 codes identified in the PCORDS table and the PCORDS table is dynamically updated to reflect the likelihood that the Type 2 codes actually identified as being present in the received input data string are likely to occur in future input data strings.[16]

De Maine describes a dynamic PCORDS table, containing multi-byte patterns for use in Type 1 compression and Type 2 codes where both the multi-byte patterns and the Type 2 codes are continuously updated based on the characteristics of each input data string received for processing. Conversely, the control code index, as recited by the claims at issue, includes control codes which are defined prior to even receiving an input data string. Furthermore, the values of the control codes in the control code index are selected to be independent of input data string specific characteristics. It is well established that each word of every claim must be given weight. See In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Likewise, Cellier does not remedy the deficiency discussed above, as Cellier describes selecting a best table of Huffman codes on the basis of a minimum cost search. In other words, the specific Huffman coding is selected based upon which code will yield the most compact encoded representation.[17] Therefore, neither De Maine, nor Cellier, alone, or in combination, disclose, or suggest, all of the features of Applicant's claims.

---

[16] De Maine at column 92, lines 46-50.
[17] Cellier at column 4, lines 46-64.

Consequently, Applicant respectfully submit that the rejection of Claims 1, 3, 5, 8-10, 21-23, 25, 26, 29-40, 44-55, and 59-62, as being unpatentable over DeMaine, and further in view of Cellier under 35 U.S.C. §103(a), is clearly in error and should be reversed.

Third Issue

The July 20, 2006 Official Action rejects Claims 6, 7, 27, and 28 as being unpatentable over DeMaine and Cellier as applied to Claims 5 and 26, respectively, and further in view Shimizu under 35 U.S.C. §103(a). The Official Action contends that De Maine and Cellier disclose all of the Applicant's claim limitations, with the exception of generating a random block size. However, the Official Action cites Shimizu as disclosing this more detailed aspect of the Applicant's invention and states that it would have been obvious to one skilled in the art at the time the invention was made to combine the cited references for arriving at the Applicant's claims. Applicant respectfully appeals this rejection.

As neither De Maine, nor Cellier, alone, or in combination, disclose all of the features of the Applicant's amended claims, and as Shimizu does not remedy the deficiency discussed above, Applicant respectfully submits that a *prima facie* case of obviousness has not been presented, and, therefore, is clearly in error and should be reversed.

Fourth Issue

The July 20, 2006 Official Action rejects Claims 41, 42, 56, and 57 are unpatentable over DeMaine and Cellier as applied to Claim 1, and further in view of Weiss under 35 U.S.C. §103(a). The Official Action contends that De Maine and Cellier disclose all of the Applicant's claim limitations, with the exception of XORing coded data. However, the

Official Action cites Weiss as disclosing this more detailed aspect of the Applicant's invention and states that it would have been obvious to one skilled in the art at the time the invention was made to combine the cited references for arriving at the Applicant's claims. Applicant respectfully appeals this rejection.

As neither De Maine, nor Cellier, alone, or in combination, disclose all of the features of the Applicant's amended claims, and as Weiss does not remedy the deficiency discussed above, Applicant respectfully submits that a *prima facie* case of obviousness has not been presented and, therefore, this rejection is clearly in error and should be reversed.

Fifth Issue

The July 20, 2006 Official Action rejects Claims 41, 43, 56, and 58 are unpatentable over DeMaine and Cellier as applied to Claim 1, and further in view of Butler under 35 U.S.C. §103(a). Applicant respectfully appeals this rejection.

As neither De Maine, nor Cellier, alone, or in combination, disclose all of the features of the Applicant's amended claims, and as Butler does not remedy the deficiency discussed above, Applicant respectfully submits that a *prima facie* case of obviousness has not been presented and, therefore, this rejection is clearly in error and should be reversed.

IX. CONCLUSION

It is believed to be clear that the outstanding rejection of July 20, 2006 fails to properly reject claims 1, 3, 5-10, 21-23, and 25-62 under 35 U.S.C. § 112, first paragraph as the required factual findings and required explanations relative to one of skill in the art have not been provided. Further, the outstanding rejection of July 20, 2006 fails to consider and/or identify all of the elements of the pending claims under 35 U.S.C. § 103 with respect tot the

rejection of claims and a reversal of the Examiner's decision is respectfully urged to be in order.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Bradley D. Lytle
Attorney of Record
Registration No. 40,073

Scott A. McKeown
Registration No. 42,866

Customer Number
22850

Tel: (703) 413-3000
Fax: (703) 413 -2220
(OSMMN 06/04)

APPENDIX I

CLAIMS ON APPEAL

Claim 1: A method of encrypting an input data string including a plurality of bits of binary data with a processing device communicatively coupled to a memory having an encryption program stored therein, the method comprising:

receiving an input data string for encryption at the processing device;

providing a control code index in the memory, the control code index being defined prior to receiving the input data string for encryption at the processing device, the control code index including a plurality of control codes;

determining an order in which to query the presence of each of 2^n different configurations of n bits within an input data string;

generating a control code associated with the determined order using the control code index, the values of the generated control code being independent of input data string specific characteristics;

generating a position code by identifying positions of each of the 2^n different configurations of n bits in the input data string in accordance with the determined order; and

combining the control code and the position code to form an encrypted data string.

Claim 3: The method of Claim 1, wherein determining an order comprises selecting a predetermined order.

Claim 5: The method of Claim 1, further comprising:

dividing the input data string into a plurality of blocks of data.

Claim 6: The method of Claim 5, wherein the number of bits within each of the plurality of blocks of data is individually determined in response to a random number generator.

Claim 7: The method of Claim 5, wherein the number of bits within each of the plurality of blocks of data is individually determined in response to a mathematical formula.

Claim 8: The method of Claim 5, further comprising: generating a plurality of block codes associated with a plurality of blocks of data of the input data string, each block code indicating the number of bits within the associated block of data.

Claim 9: The method of Claim 8, further comprising: combining the each of the plurality of block codes with the control code and the position code for the associated block of data.

Claim 10: The method of Claim 1, wherein determining an order comprises determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string.

Claim 21: A method for encrypting an input data string, including a plurality of bits of binary data, the method comprising:

receiving an input data string for encryption;

providing a control code index, the control code index being defined prior to receiving the input data string for encryption, the control code index including a plurality of control codes;

determining an order in which to query the presence of each of 2^n different configurations of n bits within an input data string;

generating a control code associated with the determined order using the control code index, the values of the generated control code being independent of input data string specific characteristics;

generating a position code by identifying positions of each of the 2^n different configurations of n bits in an input data string in accordance with the determined order; and

combining the control code and the position code to form an encrypted data string.

Claim 22: The method of Claim 21, further comprising:

arranging the input data string into a plurality of data blocks.

Claim 23: A computer readable medium including computer program instructions that cause a computer to implement a method of encrypting an input data string, including a plurality of bits of binary data, the method comprising:

receiving an input data string for encryption;

providing a control code index that is defined prior to receiving the input data string for encryption, the control code index including a plurality of control codes;

determining an order in which to query the presence of each of 2^n different configurations of n bits within an input data string;

generating a control code associated with the determined order using the control code index, the values of the generated control code being independent of input string specific characteristics;

generating a position code by identifying the positions of each of the 2^n different configurations of n bits in the input data string in accordance with the determined order; and

combining the control code and the position code to form an encrypted data string.

Claim 25: The method of Claim 23, wherein determining an order includes selecting a predetermined order.

Claim 26: The method of Claim 23, further comprising:

dividing the input data string into a plurality of blocks of data.

Claim 27: The method of Claim 26, wherein dividing the input data string into a plurality of blocks of data includes determining the individual number of bits within each of the plurality of blocks of data in response to a random number generator.

Claim 28: The method of Claim 26, wherein dividing the input data string into a plurality of blocks of data, includes determining the individual number of bits within each of the plurality of blocks of data in response to a mathematical formula.

Claim 29: The method of Claim 26, wherein determining an order further comprises:

determining a first order associated with a first block of data and determining a second order associated with a second block of data wherein the first order is different than the second order.

Claim 30: The method of Claim 26, further comprising:

generating a plurality of block codes associated with a plurality of blocks of data, each block code indicating the number of bits within the associated block of data.

Claim 31: The method of Claim 30, further comprising:

combining the each of the plurality of block codes with the control code and the position code for the associated block of data.

Claim 32: The method of Claim 23, wherein determining an order includes determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string.

Claim 33: The method of Claim 23, wherein determining an order includes determining an order in which to query the presence of each of 2^n different configurations of n bits based on an analysis of the input data string.

Claim 34: The method of Claim 23, wherein generating the control code includes generating a control code via a random number generator using the control code index.

Claim 35: The method of Claim 23, wherein determining an order includes generating an order using a mathematical formula.

Claim 36: The method of Claim 23, further comprising:

determining whether the input data string can be compressed simultaneously as it is encrypted.

Claim 37: The method of Claim 23, further comprising:

dividing the input data string into n bit sequences;

comparing each of the 2^n different configurations of n bits with each of the n bit sequences;

determining the frequency of each of the 2^n different configurations appearing in the input data string;

determining whether a specific relationship exists between values of the frequencies of each of the individual 2^n different configurations appearing in the input data string wherein the existence of the specific relationship is indicative of the presence of a characteristic within the input data string and wherein the presence of the characteristic indicates that the input data string can be compressed simultaneously as it is encrypted;

selecting a first position code routine associated with the determined order when the specific relationship exists, the first position code being operable to simultaneously encrypt and compress the input data string; and

selecting a second position code routine associated with the determined order when the specific relationship does not exist, the second position code being operable to encrypt the input data string without any compression.

Claim 38: The method of Claim 23, wherein determining the order in which to query the presence of each of 2^n different configurations of n bits within an input data string includes determining the order in which to query the presence of each of 2^2 different configurations of 2 bits within an input data string.

Claim 39: The method of Claim 38, further comprising:

dividing the input data string into n bit sequences;

comparing each of the 2^n different configurations of n bits with each of the n bit sequences of the input data string;

determining a first number representative of the number of times the most frequency occurring 2^n configuration appears in the input string.

determining a second number representative of the number of times the second most frequency occurring 2^n configuration appears in the input string;

determining a third number representative of the number of times the third most frequency occurring 2^n configuration appears in the input string;

determining a fourth number representative of the number of times the fourth most frequency occurring 2^n configuration appears in the input string;

selecting a first position code routine associated with the determined order when the first number is greater than the sum of the third number and the fourth number thereby indicating the presence of a characteristic that indicates that the input data string can be simultaneously encrypted and compressed, the first position code routine being operable to simultaneously encrypt and compress the input data string; and

selecting a second position code routine associated with the determined order when the first number is not greater than the sum of the third number and the fourth number thereby indicating the absence of the characteristic that indicates that the input data string can be simultaneously encrypted and compressed, the second position code routine being operable to encrypt the input data string without any compression.

Claim 40: The method of Claim 39, wherein generating a control code associated with the determined order, further comprises:

generating a first control code associated with the determined order when the first position code routine is selected; and

generating a second control code associated with the determined order when the second position code routine is selected wherein the first control code is different than the second control code.

Claim 41: The method of Claim 23, further comprising encrypting the encrypted data string.

Claim 42: The method of Claim 41, wherein encrypting the encrypted data string comprises:

providing an encryption key having a first selected number of bits; and

performing an XOR function between the encryption key and the encrypted data string.

Claim 43: The method of Claim 41, wherein encrypting the encrypted data string comprises:

determining an order in which to query the presence of each of 2^n different configuration of n bits with the encrypted data string;

generating a control code associated with the determined order of the encrypted data string;

generating a position code by identifying the positions of each of the 2^n different configurations o n bits in the encrypted data string in accordance with the determined order; and

combining the newly generated position code and the newly generated control code to create and encrypted version of the encrypted data string.

Claim 44: The method of Claim 25, wherein selecting a predetermined order includes computer readable code for selecting a default order.

Claim 45: The method of Claim 32, wherein determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string includes determining an order based on the relative frequencies of the 2^n combinations of the n bits of the input data string.

Claim 46: The method of Claim 32, wherein determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string includes determining a pre-determined order based on the frequencies of the 2^n combinations of the n bits of the input data string.

Claim 47: The method of Claim 1, wherein determining an order includes determining an order in which 2^n different configurations of n bits are to be identified in a position code based on an analysis of the input data string.

Claim 48: The method of Claim 1, wherein generating the control code includes generating a control code via a random number generator using the control code index.

Claim 49: The method of Claim 1, wherein determining an order includes generating an order using a mathematical formula.

Claim 50: The method of Claim 5, wherein determining an order includes determining a first order associated with a first block of data and determining a second order associated with a second block of data wherein the first order is different than the second order.

Claim 51: The method of Claim 1, further comprising:

determining whether the input data string can be compressed simultaneously as it is encrypted.

Claim 52: The method of Claim 1, further comprising:

dividing the input string into n bit sequences;

comparing each of the 2^n different configurations of n bits with each of the n bit sequences;

determining the frequency of each of the 2^n different configurations appearing in the input data string;

determining whether a specific relationship exists between values of the frequencies of each of the individual 2^n different configurations appearing in the input [[date]] data string wherein the existence of the specific relationship is indicative of the presence of a characteristic within the input data string and wherein the presence of the characteristic indicates that the input data string can be compressed simultaneously as it is encrypted;

selecting a first position code routine associated with the determined order when the specific relationship exists, the first position code being operable to simultaneously encrypt and compress the input data strong; and

selecting a second position code routine associated with the determined order when the specific relationship does not exist, the second position code being operable to encrypt the input data string without any compression.

Claim 53: The method of Claim 1, wherein determining the order in which to query the presence of each of 2^n different configurations of n bits within an input data string ~~comprises~~ includes determining the order in which to query the presence of each of 2^2 different configurations of 2 bits within an input data string.

Claim 54: The method of Claim 53, further comprising:

dividing the input data string into n bit sequences;

comparing each of the 2^n different configuration of n bits with each of the n bit sequences of the input data string;

determining a first number representative of the number of times the most frequency occurring 2^n configuration appears in the input string;

determining a second number representative of the number of times the second most frequency occurring 2^n configuration appears in the input string;

determining a third number representative of the number of times the third most frequently occurring 2^n configuration appears in the input string;

determining a fourth number representative of the number of times the fourth most frequency occurring 2^n configuration appears in the input string;

selecting a first position code routine associated with the determined order when the first number is greater than the sum of the third number and the fourth number thereby indicating the presence of a characteristic that indicates that the input data string can be simultaneously encrypted and compressed, the first position code routine being operable to simultaneously encrypt and compress the input data string; and

selecting a second position code routine associated with the determined order when the first number is not greater than the sum of the third number and the fourth number thereby indicating the absence of a characteristic that indicates that the input data string can be simultaneously encrypted and compressed, the second position code routine being operable to encrypt the input data string without any compression.

Claim 55: The method of Claim 54, wherein generating a control code associated with the determined order, further comprises:

generating a first control code associated with the determined order when the first position code routine is selected; and

generating a second control code associated with the determined order when the second position code routine is selected wherein the first control code is different than the second control code.

Claim 56: The method of Claim 1, further comprising: encrypting the encrypted data string.

Claim 57: The method of Claim 56, wherein encrypting the encrypted data string, further comprising:

providing an encryption key having a first selected number of bits; and

performing an XOR function between the encryption key and the encrypted data string.

Claim 58: The method of Claim 56, wherein encrypting the encrypted data, further comprising:

determining an order in which to query the presence of each of 2^n different configurations of n bits within the encrypted data string;

generating a control code associated with the determined order for the encrypted data string;

generating a position code by identifying positions of each of the 2^n different configurations of n bits in the encrypted data string in accordance with the determined order; and

combining the newly generated position code and the newly generated control code to create an encrypted version of the encrypted data string.

Claim 59: The method of Claim 3, wherein selecting a predetermined order includes selecting a default order.

Claim 60: The method of Claim 10, wherein determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string includes determining an order based on the relative frequencies of the 2^n combinations of the n bits of the input data string.

Claim 61: The method of Claim 10, wherein determining an order based on the frequencies of the 2^n combinations of the n bits of the input data string includes determining a pre-determined order based on the frequencies of the 2^n combinations of the n bits of the input data string.

Claim 62: An electronic device for encrypting an input data string, including a plurality of bits of binary data, comprising:

a processor configured to receive an input data string for encryption;

a memory configured to include a control code index, the control code index being defined prior to reception of the input data string for encryption by the processor, the control code index including a plurality of control codes, the control codes having corresponding values,

wherein the processor is operably linked to the memory for determining upon reception of the input data string, an order in which to query the presence of each of two 2^n different configurations of n bits within an input data string, and generates a control code

associated with the determined order by access of the control code index in which the corresponding values of the generated control code is independent of the input data string characteristics, the processor generating a position code, through the identification of positions of each of the two 2^n different configurations of n bits in the input data string in accordance with the determined order to combine the control code and the position code to form an encrypted data string.

APPENDIX II

EVIDENCE

(NONE)

APPENDIX III

RELATED PROCEEDINGS

(NONE)